STAT 401: R example for sum of squares

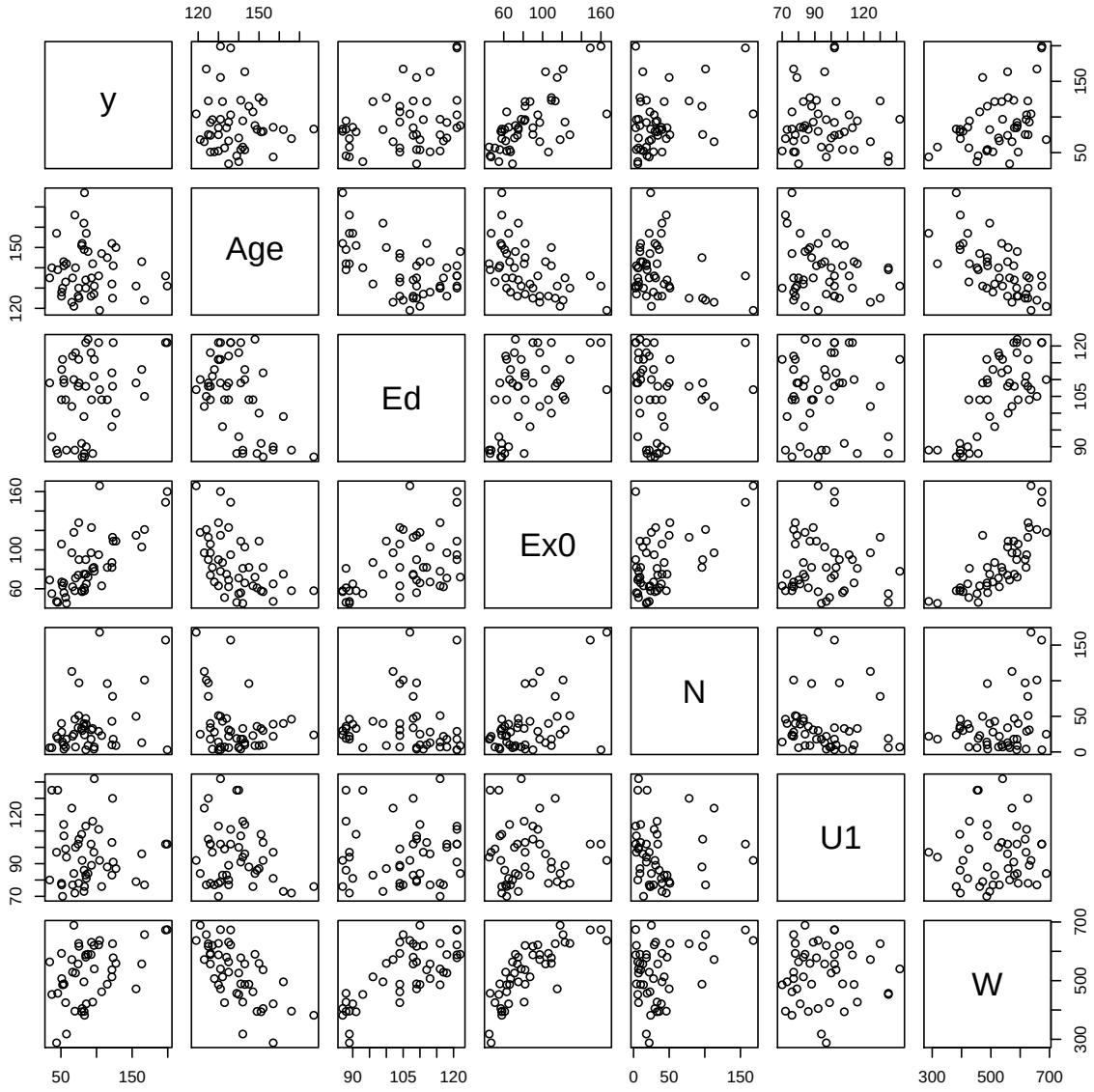
Zepu Zhang

November 4, 2010

I found a US crime dataset on the internet (likely similar to one of the datasets mentioned in the textbook). Description is attached.

```
> data <- read.table('USCrime.txt', header = TRUE)
>
> print(names(data))
 [1] "R"   "Age" "S"   "Ed"  "Ex0" "Ex1" "LF"  "M"   "N"   "NW"  "U1"  "U2"
[13] "W"   "X"
>
> # 'R' is the reponse (crime rate)
> # Since we know some predictors are highly correlated,
> # and we don't deal with that problem now,
> # we'll focus on the following predictors:
> #  Age: number of male aged 14--24 per 1000 population
> #  Ed:  mean # of years of schooling
> #  Ex0: per capita expenditure on police by government
> #  N:   state population
> #  U1:  unemployment rate of urban males
> #  W:   median family goods
>
> y <- data$R
> X <- as.matrix(data[, c('Age', 'Ed', 'Ex0', 'N', 'U1', 'W')])
> X <- cbind(1, X)   # Add the constant predictor.
> colnames(X)[1] <- 'Const'
> n <- length(y)
> p <- ncol(X)
> alpha <- .05
>

> ##### BLOCK 1 #####
>
> # Take a look at the data.
> pdf(file = 'part9.scm.pdf', width = 8, height = 8)
> pairs(cbind(y, X[, -1]))
> dev.off()
null device
          1
```




```
>
>
> print(sstotal <- sum(y * y))
[1] 453823.4
> print(syy <- sstotal.corrected <- sum((y - mean(y))^2))
[1] 68809.28
>     # Does the assignment and printing on one line, to be lazy.
>     # 'print' applies on 'syy'.
>     # Compare sst and syy!
>
```

```
>
> ##### BLOCK 2 #####
>
> # Let's fit a model with the intercept only.
> z <- lm.fit(x = X[, 'Const', drop = FALSE], y = y)
>     # Selecting a single row or col will return a simple vector
>     # by default; use 'drop = FALSE' will keep it a matrix.
>     # 'drop' means dropping the dimension info.
> print(coef(z))
   Const
90.50851
>
> # We know this estimate should equal the mean.
> # Does it?
> print(mean(y))
[1] 90.50851
>
> # Calc SSR and SSE.
> print(ssr.Const <- sum(fitted(z) ^2))    # SSR
[1] 385014.2
> print(sse.Const <- sum(residuals(z) ^2)) # SSE.
[1] 68809.28
>     # This should equal syy. Does it?
> print(syy)
[1] 68809.28
> print(ssr.Const + sse.Const)
[1] 453823.4
>     # This should equal sstotal. Does it?
> print(sstotal)
[1] 453823.4
>
> # Let's test the significance of this 'pure-intercept' model.
> print((ssr.Const / 1) / (sse.Const / (n - 1)))
[1] 257.3875
> print(qf(1 - alpha, 1, n - 1))
[1] 4.051749
>     # Is the test statistic greater than the critical value?
>     # Is it surprising to you that the intercept is significant?
>

>
>
> ##### BLOCK 3 #####
>
> # Let's add predictor 'Age'.
```

```
> z <- lm.fit(x = X[, c('Const', 'Age')], y = y)
> print(coef(z))
      Const         Age
128.6645573  -0.2753469
>
> # Calc SSR and SSE.
> print(ssr.ConstAge <- sum(fitted(z) ^2))    # SSR
[1] 385565
> print(sse.ConstAge <- sum(residuals(z) ^2)) # SSE.
[1] 68258.44
> print(ssr.ConstAge + sse.ConstAge)
[1] 453823.4
>     # This should equal sstotal. Does it?
> print(sstotal)
[1] 453823.4
>
> # Let's test the significance of the coef for 'Age'.
> print(((ssr.ConstAge - ssr.Const)/ 1) / (sse.ConstAge / (n - 2)))
[1] 0.3631461
> print(qf(1 - alpha, 1, n - 2))
[1] 4.056612
>     # Is the test statistic greater than the critical value?
>
> # Turns out to be insignificant.
> # Compare the sse's:
> print(sse.Const)
[1] 68809.28
> print(sse.ConstAge)
[1] 68258.44
>     # The decrease is indeed small.
>
> # ssr.ConstAge - ssr.Const should be equal to
> # sse.Const - sse.ConstAge.
> # Is it?
> print(ssr.ConstAge - ssr.Const)
[1] 550.8396
> print(sse.Const - sse.ConstAge)
[1] 550.8396
>     # Both are SSR(Age | Const)
>
> # Out of curiosity,
> # is the extra SS of 'Age' the same as the SSR of 'Age' alone?
> z <- lm.fit(x = X[, 'Age', drop = FALSE], y = y)
> print(ssr.Age <- sum(fitted(z) ^ 2))
```

```
[1] 379351.5
>
> # Compare the above with SSR(Age | Const).
> # The SSR of 'Age' alone is much larger than its extra contribution
> # on top of 'Const'.
>
> # Now, adding 'Age' on top of 'Const' is not significant.
> # Does 'Age' alone makes a significant model?
> sse.Age <- sum(residuals(z) ^ 2)
> print( (ssr.Age / 1) / (sse.Age / (n - 1)) )
[1] 234.3188
> print(qf(1 - alpha, 1, n - 1))
[1] 4.051749
>     # Is the test statistic greater than the critical value?
>

>
> ##### BLOCK 4 #####
>
> # Does 'Const' and 'Age' as a group make a significant model?
> # The answer must be 'yes', given the preceding results.
> # But let's do a test anyway.
> # The things we need are already computed.
> print( (ssr.ConstAge / 2) / (sse.ConstAge / (n - 2)) )
[1] 127.0936
> print(qf(1 - alpha, 2, n - 2))
[1] 3.204317
>

> ##### BLOCK 5 #####
>
> # Keeping 'Const' and 'Age' in the model,
> # let's add 'Ed' and 'Ex0' at once.
> z <- lm.fit(x = X[, c('Const', 'Age', 'Ed', 'Ex0')], y = y)
> print(coef(z))
      Const           Age            Ed           Ex0
-221.0878571    1.2300299    0.4737244    1.0717904
>
> ssr.CAEE <- sum(fitted(z) ^ 2)
> sse.CAEE <- sum(residuals(z) ^ 2)
> print(ssr.CAEE + sse.CAEE)  # This should be equal to 'sstotal'.
[1] 453823.4
> print(sstotal)
[1] 453823.4
>
```

```
> # Let's test the group.
> print( ((ssr.CAEE - ssr.ConstAge) / 2) / (sse.CAEE / (n - 4)) )
[1] 28.65531
> print(qf(1 - alpha, 2, n - 4))
[1] 3.214480
>
>
```

The Data and Story Library

DASL (pronounced "dazzle") is an online library of datafiles and stories that illustrate the use of basic statistics methods. We hope to provide data from a wide variety of topics so that statistics teachers can find real-world examples that will be interesting to their students. Use DASL's powerful search engine to locate the story or datafile of interest.

Power Search

List all topics | List all methods | Data subjects

Help! | Submit a story | © Copyright

Overview

Teachers use examples to illustrate statistics concepts. A good example can make a lesson on a particular statistics method vivid and relevant. DASL is designed to help teachers locate and identify datafiles for teaching. We hope that DASL will also serve as an archive for datasets from the statistics literature.

The archive contains two types of files, stories and datafiles. Each story applies a particular statistical method to a set of data. Each datafile has one or more associated stories. The data can be downloaded as a space- or tab-delimited table of text, easily read by most statistics programs.

Stories are classified according to statistical methods and major topics of interest. Power search through DASL's stories and datafiles in five different ways.

1. **Title Search:** Searches through all of the story titles.
2. **Method Search:** Statistical methods such as regression or ANOVA.
3. **Topic Search:** Topics such as psychology or health.
4. **Datafile Subject Search:** Data subjects such as finance or astronomy.
5. **Full-text Search:** Searches through all of the stories and datafiles.

The first four specialized searches are slightly faster than the full-text search. Use these searches if you know what you want. The full-text search is helpful if you're interested in something more general (e.g., Fisher). Help Search provides information on using the search engines and provides a few examples.

Final Comments & Thanks

DASL is part of larger effort to enhance the teaching of statistics using computers. A related project, the Electronic Encyclopedia of Statistical Exercises and Examples (EESEE), offers a self-study application. Another wonderful location to visit is the Chance Database. Chance also provides a link to several other Statistics Related internet sources such as the American Statistical Association, International Association for Statistical Computing and more.

The hard disk storage space for DASL is generously provided by StatLib.

Thank you for visiting DASL. Please send us your comments and suggestions.

Main Menu | Power Search
List all topics | List all methods | List all datafile subjects
DASL Help | Submit your story | Copyright

Story Name:
US Crime
Story Topics:
Social science
Datafile Name:
US Crime
Methods:
Collinearity , Correlation , Causation , Lurking variable , Regression
Abstract:
These data are crime-related and demographic statistics for 47 US states in 1960. The data were collected from the FBI's *Uniform Crime Report* and other government agencies to determine how the dependent variable crime rate (R) depends on the other variables measured in the study.

We encounter many problems analyzing these data by regression because some predictor variables are highly correlated. For example, Ex0 and Ex1, which measure police expenditures in consecutive years, have a correlation of .99. Wealth (W) and income inequality (X) are also highly correlated, as are U1 and U2, which measure unemployment in two different age groups. When predictor variables are highly correlated, the model is said to be nearly collinear. The result is that our estimated coefficients are unstable; removing one variable from the model may cause the results for the other variables to change dramatically.

In addition, the causal relationship between Ex0 (expenditures in 1960) and crime rate is unclear. Do increased expenditures affect the crime rate, or does the crime rate motivate an increase in expenditures?

In one possible analysis, predictors are removed from the model until only the 5% significant predictors Age, Ed, U2, X, and Ex0 remain. The results of this model are in Figure 1. This model demonstrates that it is important to look at the direction of the coefficients. From these coefficients, it appears that more education and police expenditures increase the crime rate. Perhaps there is another variable, a "lurking variable" not collected with these data, which causes both education and crime rate to increase together.

This data set is a good example of what can go wrong in a regression analysis.
Image:
Results for a possible model for these data

Dependent variable is: **R**
No Selector
48 total cases of which 1 is missing
R squared = 73.0% R squared (adjusted) = 69.7%
s = 21.30 with 47 - 6 = 41 degrees of freedom

Source	Sum of Squares	df	Mean Square	F-ratio
Regression	50205.6	5	10041.1	22.1
Residual	18603.6	41	453.747	

Variable	Coefficient	s.e. of Coeff	t-ratio	prob
Constant	-524.374	95.12	-5.51	≤ 0.0001
Age	1.01982	0.3532	2.89	0.0062
Ed	2.03077	0.4742	4.28	0.0001
U2	0.913608	0.4341	2.10	0.0415
X	0.634926	0.1468	4.32	≤ 0.0001
Ex0	1.23312	0.1416	8.71	≤ 0.0001

Go Main Menu | Power Search | Data subjects

Datafile Name:
US Crime
Datafile Subjects:
Social science
Story Names:
US Crime
Reference:
Vandaele, W. (1978) Participation in illegitimate activities: Erlich revisited. In *Deterrence and incapacitation*, Blumstein, A., Cohen, J. and Nagin, D., eds., Washington, D.C.: National Academy of Sciences, 270-335. Methods: A Primer, New York: Chapman & Hall, 11. Also found in: Hand, D.J., *et al.* (1994) *A Handbook of Small Data Sets*, London: Chapman & Hall, 101-103.
Authorization:
Contact author
Description:
These data are crime-related and demographic statistics for 47 US states in 1960. The data were collected from the FBI's *Uniform Crime Report* and other government agencies to determine how the variable crime rate depends on the other variables measured in the study.
Number of cases:
47
Variable Names:

1. R: Crime rate: # of offenses reported to police per million population
2. Age: The number of males of age 14-24 per 1000 population
3. S: Indicator variable for Southern states (0 = No, 1 = Yes)
4. Ed: Mean # of years of schooling x 10 for persons of age 25 or older
5. Ex0: 1960 per capita expenditure on police by state and local government
6. Ex1: 1959 per capita expenditure on police by state and local government
7. LF: Labor force participation rate per 1000 civilian urban males age 14-24
8. M: The number of males per 1000 females
9. N: State population size in hundred thousands
10. NW: The number of non-whites per 1000 population
11. U1: Unemployment rate of urban males per 1000 of age 14-24
12. U2: Unemployment rate of urban males per 1000 of age 35-39
13. W: Median value of transferable goods and assets or family income in tens of $
14. X: The number of families per 1000 earning below 1/2 the median income

The Data:

R	Age	S	Ed	Ex0	Ex1	LF	M	N	NW	U1	U2	W	X
79.1	151	1	91	58	56	510	950	33	301	108	41	394	261
163.5	143	0	113	103	95	583	1012	13	102	96	36	557	194
57.8	142	1	89	45	44	533	969	18	219	94	33	318	250
196.9	136	0	121	149	141	577	994	157	80	102	39	673	167

123.4	141	0	121	109	101	591	985	18	30	91	20	578	174
68.2	121	0	110	118	115	547	964	25	44	84	29	689	126
96.3	127	1	111	82	79	519	982	4	139	97	38	620	168
155.5	131	1	109	115	109	542	969	50	179	79	35	472	206
85.6	157	1	90	65	62	553	955	39	286	81	28	421	239
70.5	140	0	118	71	68	632	1029	7	15	100	24	526	174
167.4	124	0	105	121	116	580	966	101	106	77	35	657	170
84.9	134	0	108	75	71	595	972	47	59	83	31	580	172
51.1	128	0	113	67	60	624	972	28	10	77	25	507	206
66.4	135	0	117	62	61	595	986	22	46	77	27	529	190
79.8	152	1	87	57	53	530	986	30	72	92	43	405	264
94.6	142	1	88	81	77	497	956	33	321	116	47	427	247
53.9	143	0	110	66	63	537	977	10	6	114	35	487	166
92.9	135	1	104	123	115	537	978	31	170	89	34	631	165
75.0	130	0	116	128	128	536	934	51	24	78	34	627	135
122.5	125	0	108	113	105	567	985	78	94	130	58	626	166
74.2	126	0	108	74	67	602	984	34	12	102	33	557	195
43.9	157	1	89	47	44	512	962	22	423	97	34	288	276
121.6	132	0	96	87	83	564	953	43	92	83	32	513	227
96.8	131	0	116	78	73	574	1038	7	36	142	42	540	176
52.3	130	0	116	63	57	641	984	14	26	70	21	486	196
199.3	131	0	121	160	143	631	1071	3	77	102	41	674	152
34.2	135	0	109	69	71	540	965	6	4	80	22	564	139
121.6	152	0	112	82	76	571	1018	10	79	103	28	537	215
104.3	119	0	107	166	157	521	938	168	89	92	36	637	154
69.6	166	1	89	58	54	521	973	46	254	72	26	396	237
37.3	140	0	93	55	54	535	1045	6	20	135	40	453	200
75.4	125	0	109	90	81	586	964	97	82	105	43	617	163
107.2	147	1	104	63	64	560	972	23	95	76	24	462	233
92.3	126	0	118	97	97	542	990	18	21	102	35	589	166
65.3	123	0	102	97	87	526	948	113	76	124	50	572	158
127.2	150	0	100	109	98	531	964	9	24	87	38	559	153
83.1	177	1	87	58	56	638	974	24	349	76	28	382	254
56.6	133	0	104	51	47	599	1024	7	40	99	27	425	225
82.6	149	1	88	61	54	515	953	36	165	86	35	395	251
115.1	145	1	104	82	74	560	981	96	126	88	31	488	228
88.0	148	0	122	72	66	601	998	9	19	84	20	590	144
54.2	141	0	109	56	54	523	968	4	2	107	37	489	170
82.3	162	1	99	75	70	522	996	40	208	73	27	496	224
103.0	136	0	121	95	96	574	1012	29	36	111	37	622	162
45.5	139	1	88	46	41	480	968	19	49	135	53	457	249
50.8	126	0	104	106	97	599	989	40	24	78	25	593	171
84.9	130	0	121	90	91	623	1049	3	22	113	40	588	160